# Statistical modelling of spatio-temporal dependencies in NGS data

Ranciati, Saverio

# Chapter 4

# Bayesian Smooth-and-Match estimation of ODEs' parameters with quantifiable solution uncertainty

**Saverio Ranciati**[1,2], **Cinzia Viroli**[2], **Ernst C. Wit**[1]

[1] *University of Groningen, Johann Bernoulli Institute of Mathematics and Computer Science, Groningen, the Netherlands.*
[2] *Department of Statistics, University of Bologna, Bologna, Italy*

## Abstract

In many fields of application, dynamic processes that evolve through time are well described by systems of ordinary differential equations (ODEs). The parameters governing the equations cannot be immediately extrapolated out of the data when the analytical solution of the ODEs is not available. Different methods have been proposed to infer these quantities, from numerical optimization to more general models that employs regularization as a way to overcome the issue of trying to directly solve the system. We focus on the class of techniques that use smoothing to avoid direct integration and, in particular, on a Bayesian Smooth-and-Match strategy that allows to obtain the ODEs solution while performing inference on the parameters, with a built-in quantification of the related uncertainty. We assess the performance of the proposed approach in three different simulation studies and we compare the results on a dataset on neuron electrical activity.



**Keywords:** ordinary differential equations; smoothing; penalized cubic splines; MCMC; ridge regression.

## 4.1 Introduction

Many processes that evolve through time are described by systems of ordinary differential equations (ODEs). Imagine we have a simple case of one ODE where the change, $dx(t)$, of the concentration level of a specific molecule in the cell follows a law that is described by some function $g_{\beta}[x(t)]$, with a set of parameter $\boldsymbol{\beta}$ governing this law. We could think of this change as depending on the quantity of molecules at time $t$, that is $x(t)$, a rate $\beta_1$ at which new ones are produced by the cell as time passes by and then maybe some 'limit' $\beta_2$ on the capacity to contain them. What we observe, in practice, are not the actual changes $dx(t)$ (the derivative of the process) but instead the concentration levels $x(t)$ at sampling time points. This means that, in order to relate the data at our disposal with the parameters of interest, we need a solution for the system of differential equations. Solving the problem analytically however is not always possible and in that case no closed forms are available for the estimation of the parameters. Moreover, our observations may very well be affected by noise that perturbs the true temporal dynamic of the process. There are several techniques in the literature on this topic (see Robinson [25] for an introduction): most of them involve numerical integration, a straightforward approach to the problem that, however, does not take into account in any way the uncertainty about the solution of the system. Moreover, methods relying on this type of solvers require an explicit computation of the solution at every step of the algorithm, severely hindering the procedure in practice. A way to avoid direct numerical integration (or differentiation) is *smoothing* the data. This idea falls under the class of *collocation* methods, in which some of them are called *two-steps* [9, 15]. As in Brunel et al. [4], Madár et al. [20], Varah [27], a first step consist of recovering a temporary solution of the system by smoothing or interpolating the data (i.e. with cubic splines, least squares, nonparametric filters, local polynomial regression and so forth) and then applying nonlinear least squares to infer the parameters of the ODEs. The properties of these methods, such as consistency and asymptotic normality, are discussed in Xue et al. [30]. Other methods following a similar strategy of smoothing and matching are discussed in Campbell and Steele [6], González et al. [13], Liang and Wu [18]. Another approach is to use regularization [14, 24, 28], in order to do infer-

ence on the parameters while minimizing - with a frequentist flavor - some measure of distance between the theoretical solution and the estimated one. An initial guess of the parameters of interest is provided to the algorithm, as it is used together with a linear combination of basis functions to solve a penalized optimization problem. From a Bayesian point of view, as in Chkrebtii et al. [7], Gaussian Processes (GP) are prominent tools employed to solve the task. They encode naturally a source of randomness in the solution and simultaneously provide a class of flexible priors for the functions used to smooth the data coming from the ODEs' system. A recent approach using GP is provided by Calderhead et al. [5], with some drawbacks that were later addressed in Dondelinger et al. [10] through the use of adaptive gradient matching. Another advantage of these Bayesian methods is the complete probabilistic phrasing of the problem, allowing for a statistical quantification of the uncertainty about the solution obtained: the core of these procedures are - in fact - probabilistic solvers that can be sampled to explore the parameter space while obtaining indirectly a solution of the system [8]. For some other (implicit and explicit) probabilistic solvers see Barber [1]; some applications of these methods on real life dataset are presented in Honkela et al. [16] and Titsias et al. [26].

In this work, we propose a two-step Bayesian strategy (Bayesian Smooth-and-Match) that borrows the idea of smoothing to overcome direct integration and, simultaneously, to filter some of the noise in the data. The first step of the method relies on penalized splines to smooth the data and reconstruct the variables of the ODEs; the second step focuses on inferring the parameters of the system through ridge regression, with covariates being known functions of the process that is being studied.

The rest of the manuscript is organized as follows: in Section 4.2, the notation used to build the strategy is introduced and we discuss the distributional assumptions on the data, together with the prior and posterior distributions; in Section 4.3, numerical and visual results on three different simulation studies are reported; in Section 4.3.4, an application to a previously analyzed dataset is presented; finally, in Section 4.4, we provide a summary of the work and we outline some future developments.

## 4.2 Model formulation

### 4.2.1 Tools and notation

Suppose we observe a $p$-dimensional vector $\boldsymbol{y}_i$ containing noisy observations of a dynamic process $\boldsymbol{x}(t_i)$, where $t_i$ is a generic element from the set of ordered time points $\{t_i\} \in [0, 1]$, and $i = 1, \ldots, n$ indexing the independent observed vectors. For the sake of simplicity, we will drop the "$(t)$" argument in the notation for $\boldsymbol{x}(t)$ and its components when not ambiguous, otherwise it will be explicit. We assume that:

$$y_{ik} \sim N\left(x_{ik}, \sigma_k^2\right) \tag{4.1}$$

with $k = 1, \ldots, p$, $x_{ik} = x_k(t_i)$ and $\sigma_k^2$ a parameter describing the noise level in the data for the component $k$ of the whole process $\boldsymbol{x}$. For every observation $t_i$, each component $\boldsymbol{x}_{\cdot k}$ can be approximated by a cubic spline [29, p. 122] of $t$ in the time domain

$$x_{ik} = \sum_{h=1}^{q_k+2} \theta_{hk}\psi_{hk}(t_i) \tag{4.2}$$

where the sum is over $q_k + 2$ known basis functions $\psi_{hk}(\cdot)$ that depend on the $q_k$ number of arbitrary knots used to construct them. For an arbitrary fixed $t$, we assume that the dynamic process $\boldsymbol{x}(t)$ is well described by an ordinary differential equations (ODEs) model defined as:

$$\begin{cases} \boldsymbol{x}'(t) = \frac{d\boldsymbol{x}(t)}{dt} = g(\boldsymbol{x}(t)) \\ \boldsymbol{x}(0) = \boldsymbol{\xi} \end{cases} \tag{4.3}$$

where $\boldsymbol{x}'(t)$ is the first derivative w.r.t. time of a continuous process $\boldsymbol{x}(t) = (\boldsymbol{x}_{\cdot 1}(t), \ldots, \boldsymbol{x}_{\cdot k}(t), \ldots, \boldsymbol{x}_{\cdot p}(t))$, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_k, \ldots, \xi_p)$ is a vector of initial conditions for the system and $g : \mathbb{R}^p \to \mathbb{R}^p$ a (possibly) non-linear function of $\boldsymbol{x}(t)$. We focus on ODEs models that are linear in the parameters. The generic functional form of the derivative for the first variable $\boldsymbol{x}_{\cdot 1}$ is

$$\frac{d\boldsymbol{x}_{\cdot 1}(t)}{dt} = g_1(\boldsymbol{x}(t)) = \sum_{j=1}^{b} \beta_j h_j(\boldsymbol{x}(t)) \tag{4.4}$$

which involves the first element $g_1 : \mathbb{R}^p \to \mathbb{R}$ of the function $g$. We think of it as a linear combination of $b$ parameters of interest and some functions $h_j$, with $j = 1, \ldots, b$, that describe the dynamic evolution of the component $\boldsymbol{x}_{\cdot 1}$. Instead of working with the derivative of $\boldsymbol{x}$ we switch to the integral representation of the system in Equation 4.3:

$$\begin{aligned} \boldsymbol{x}_{i1} = \int_{t_1}^{t_i} \frac{d\boldsymbol{x}_{\cdot 1}(t)}{dt} &= \int_{t_1}^{t_i} g_1(\boldsymbol{x}(s))ds = \\ &= \xi_1 + \int_{t_1}^{t_i} \sum_{j=1}^{b} \beta_j h_j(\boldsymbol{x}(s))ds \end{aligned} \tag{4.5}$$

for $i = 2, \ldots, n$, because we set our first observation $t_1$ as the starting time point ($t_1$ could be either zero or not), and $\xi_1 = x_{11}$. The initial condition $\xi_1$ can be either estimated or assumed known. The solution for the first ODE, given by Equation 4.5, is thus:

$$\boldsymbol{x}_{i1} = \xi_1 + \sum_{j=1}^{b} \left( \beta_j \int_{t_1}^{t_i} h_j(\boldsymbol{x}(s))ds \right) = \xi_1 + \sum_{j=1}^{b} \beta_j H_j(\boldsymbol{x}(t_i)) \tag{4.6}$$

where $H_j$ is the integral of the corresponding function $h_j$ evaluated at time point $t_i$.

### 4.2.2 Prior, likelihood and posterior distributions

With reference to a sample of $n$ observations, Equation (4.1) may be rewritten as:

$$\boldsymbol{y}_{\cdot k} = (y_{1k}, \ldots, y_{nk})^T \sim \mathcal{N}_n \left( \boldsymbol{x}_{\cdot k}, \sigma_k^2 \mathrm{I}_n \right)$$

with $\sigma_k^2$ the noise level for component $k$ and $\mathrm{I}_n$ the identity matrix of order $n$. From Equation 4.2, $\boldsymbol{x}_{\cdot k} = \Psi_k \boldsymbol{\theta}_k$ where $\Psi$ is the $n \times (q_k + 2)$ matrix of spline basis evaluated at every time point $t_i$. Thus, assuming that every component is independent from the other given the column vectors $\boldsymbol{\theta}_k$, the likelihood function of the model is

$$P(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{\sigma}^2) = \prod_{i=1}^{n} \prod_{k=1}^{p} P(y_{ik}|\boldsymbol{\theta}_k, \sigma_k^2)$$

where $\boldsymbol{y} = (\boldsymbol{y}_{\cdot 1}, \ldots, \boldsymbol{y}_{\cdot p})$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p\}$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_p^2)$. We choose to tackle the inferential procedure with a Bayesian ap-

proach: this allows us to represent the whole process with a fully probabilistic generative model that we can also describe as a graphical model. Furthermore, within the Bayesian framework, we can take into account the variability and the uncertainty at every level of the problem, from the smoothing of the data to the parameters governing the system. We thus assign prior probabilities to the parameters of the splines: we assume, for each $\boldsymbol{\theta}_k$, a Gaussian prior distribution of the form

$$\boldsymbol{\theta}_k | \, \lambda_{\theta_k} \sim \mathcal{N}_{q_k+2} \left( \mathbf{0}, [\lambda_{\theta_k} \mathrm{S}_{\theta_k}]^{-1} \right).$$

Choosing this prior is equivalent to performing a penalized spline regression on the data: the form of the hyperparameter precision matrix $\mathrm{S}_{\theta_k}$ defines the way we want the basis to be penalized. We follow this approach to ensure that non-linearity in the components is captured without, however, producing curves that would overfit the noisy data. The parameter $\lambda_{\theta_k}$ penalizes the non-smoothness of the functions $\psi_{hk}$: the 'wiggliness' of the curve resulting from the spline smoothing is encoded in the precision matrix $\mathrm{S}_{\theta_k}$ (see Wood [29] for details, p. 126). These penalization terms have prior distributions

$$\lambda_{\theta_k} | \, \alpha_{\theta_k}, \gamma_{\theta_k} \sim \mathrm{Gam}(\alpha_{\theta_k}, \gamma_{\theta_k})$$

for some vectors of shape and rate hyperparameters $(\boldsymbol{\alpha}_\theta, \boldsymbol{\gamma}_\theta)$. These two-dimensional vectors are chosen to represent weakly informative priors on $\lambda_{\theta_k}$: more specifically, we select values of the hyperparameters that encourage undersmoothing of the data [15], with enough variance for the Gamma distribution to be able to shift to a more penalized curve if needed. A reference improper prior density $P(\sigma_k^2) = 1/\sigma_k^2$ is employed for each $\sigma_k^2$. The first vector of observations $\boldsymbol{y}_{\cdot 1}$ has another representation, stemming directly from the ODEs' system solution for $\boldsymbol{x}_{\cdot 1}$ that depends on the integral solution from Equation 4.6, that is

$$\boldsymbol{y}_{\cdot 1}^{\star} | \, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2 \sim \mathcal{N}_n \left( \xi_1 \mathbf{1}_n + \boldsymbol{H}\boldsymbol{\beta}, \sigma_1^2 \mathrm{I}_n \right)$$

where $\mathbf{1}_n$ is a $n$-dimensional unitary column vector, $\boldsymbol{\beta}$ the column vector of parameters we want to estimate and $\boldsymbol{H}$ a $n \times b$ matrix collecting the integrated functions $H_j$ evaluated for each observation $t_i$. As in our case, when the starting inverse-problem may be 'ill-posed', sure enough the ordinary least squares estimation leads to overdetermined

or underdetermined systems of equations as solution to the regression itself. Regularization is the usual approach to overcome this issue and Tikhonov regularization, in particular, is one of the most commonly used: from a Bayesian point of view, it is equivalent to assume, for $\boldsymbol{\beta}$, the following prior distribution

$$\boldsymbol{\beta}|\lambda_\beta \sim \mathcal{N}_b\left(\mathbf{0}, [\lambda_\beta \mathrm{I}_b]^{-1}\right).$$

The choice of this prior, effectively, induces the Bayesian 'ridge regression' with $\lambda_\beta$ acting as a penalizing term for which we assume a prior distribution

$$\lambda_\beta|\alpha_\beta, \gamma_\beta \sim \mathrm{Gam}(\alpha_\beta, \gamma_\beta).$$

As for the other penalization terms, we follow the same approach of using a weakly informative prior. We also estimate $\xi_1$ instead of rescaling the data and we select a flat prior for it. Dropping hyperparameters from the notation, the joint posterior distribution of the parameters in our model is:

$$\begin{aligned} & P(\boldsymbol{\Theta}, \xi_1, \boldsymbol{\beta}, \boldsymbol{\lambda}_\theta, \lambda_\beta, \boldsymbol{\sigma}^2|\, \boldsymbol{y}, \boldsymbol{y}^\star_{\cdot 1}) \propto P\big(\boldsymbol{y}^\star_{\cdot 1}|\, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2\big) \times \\ \times & \prod_{k=1}^{p} \left[P\big(\boldsymbol{y}_{\cdot k}|\, \boldsymbol{\theta}_k, \sigma_k^2\big)\, P(\boldsymbol{\theta}_k|\, \lambda_{\theta_k})\, P(\lambda_{\theta_k}) P(\sigma_k^2)\right] \times \\ \times & P(\boldsymbol{\beta}|\, \lambda_\beta)\, P(\lambda_\beta)\, P(\xi_1) \end{aligned} \tag{4.7}$$

which is represented in the graphical model in Figure 4.1.

### 4.2.3 Mimicking the data: relationship with other methods

One feature of this representation is that the vector of observations $\boldsymbol{y}_{\cdot 1}$ appears twice: as a term of the likelihood component, when $k = 1$ in the first product, and then as the noisy solution of the ODEs' system, that is $\boldsymbol{y}^\star_{\cdot 1}$. In Barber and Wang [2], also their GPODE model focuses on a probabilistic generative model for the data and the graph representation contains two nodes for the same quantity (namely, the process itself). The authors model the system $\boldsymbol{x}(t)$ as coming from a Gaussian process (GP) and exploit the fact that differentiating the GP produces derivatives $\boldsymbol{x}'(t)$ that are still modeled as a Gaussian process with available analytical description of its kernel (see the original manuscript for more details). Then, they marginalize over the components $\boldsymbol{x}(t)$ with a standard convolution integral and model the data $\boldsymbol{y}$

with a Gaussian distribution; after that, they reintroduce $\boldsymbol{x}(t)$ and couple it with the obtained derivatives to measure the distance between the deterministic ODEs of the system and the ones estimated by the data. This approach faces some issues. As pointed out in Macdonald et al. [19], where the authors inspect the graph representation in Barber and Wang [2], having two nodes assigned to the same quantity is methodologically inconsistent. To solve the issue, they first introduce a dummy variable that mimics $\boldsymbol{x}(t)$, thus removing the inconsistency. However, the two nodes are still conceptually describing the very same quantity and a natural definition of this dependency would be an undirected edge between them: this addition, unfortunately, changes the graph from a directed acyclic graph to a chain graph which is not a probabilistic generative model anymore. To preserve its nature, they keep the two nodes separate form one another but highlight the consequences of this choice: when some of the noisy vectors $\boldsymbol{y}_{\cdot k}$ are not available (that is the case of partially observed systems), the model itself might be unidentifiable because of the likelihood not depending anymore on the parameters of the ODEs after marginalizing over the unobserved quantities. As we face the same issue with our proposed strategy, we are thus limited to situations where all the components of the process $\boldsymbol{x}(t)$ are observed (with noise). Given that a probabilistic definition of what could be the directed edge between $\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}^{\star}_{\cdot 1}$ (or viceversa) is not obvious, we couple the two quantities only by assuming they share the same variance $\sigma_1^2$ (as described in Figure 4.1). This shared nuisance parameter, however, suffers from the independence assumption on $\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}^{\star}_{\cdot 1}$. Suppose the simplest case where $\boldsymbol{y}^{\star}_{\cdot 1} = \boldsymbol{y}_{\cdot 1} \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 I_n)$: having no edge between the two nodes is equivalent to $P(\boldsymbol{y}^{\star}_{\cdot 1}, \boldsymbol{y}_{\cdot 1} | \sigma_1^2) = P(\boldsymbol{y}^{\star}_{\cdot 1} | \sigma_1^2) P(\boldsymbol{y}_{\cdot 1} | \sigma_1^2)$ which is roughly equal to a Normal distribution with variance parameter $\frac{\sigma_1^2}{2}$. Obviously, this would not happen if a directed edge were to be added to the graph, as the joint distribution of the vector and its copy would be instead $P(\boldsymbol{y}^{\star}_{\cdot 1}, \boldsymbol{y}_{\cdot 1} | \sigma_1^2) = P(\boldsymbol{y}^{\star}_{\cdot 1} | \boldsymbol{y}_{\cdot 1}, \sigma_1^2) P(\boldsymbol{y}_{\cdot 1} | \sigma_1^2) = 1 \cdot P(\boldsymbol{y}_{\cdot 1} | \sigma_1^2)$, but we already discussed there is no clear way to describe a directed edge between the two nodes. This inconvenience is mostly present in a simulation environment. We start with a deterministic solution of an ODEs' system, $\boldsymbol{x}_{\cdot 1}$, and we perturb it with some noise $\tilde{\sigma}^2$ thus obtaining an observed vector $\boldsymbol{y}_{\cdot 1}$: recovering the noise level in the data

is not equivalent anymore to estimating $\tilde{\sigma}^2$, as in our model we are bringing together in $\sigma_1^2$ two sources of noise - albeit coming from the very same vector of observations. In fact, $\sigma_1^2$ bears both the uncertainty about how good the solution of the set of ODEs and also how good is the smoothing of the first vector of data that we used as a regressor in solving indirectly the system. Such an undesired 'mismatch' effect should not present itself as a problem when dealing with real world observations.

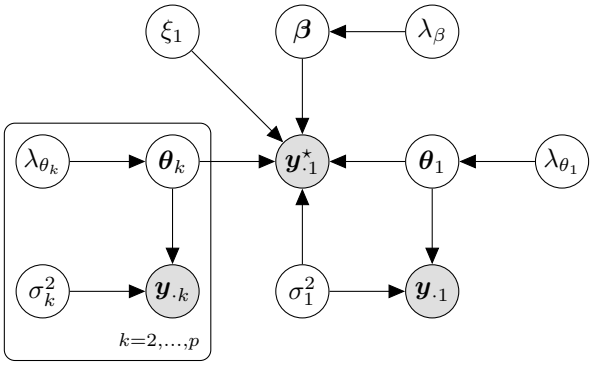


Figure 4.1: Graphical representation of the ODEs and solution model

### 4.2.4 Bayesian Smooth-and-Match

Following the standard derivations for the full conditionals from Equation 4.7, the samplers for $\boldsymbol{\theta}_k$ should consider the quantities

$$\begin{aligned} P(\boldsymbol{\theta}_k|\sigma_k^2, \lambda_{\theta_k}, \boldsymbol{y}_{\cdot k}) \;&\propto\; P\big(\boldsymbol{y}_{\cdot k}|\,\boldsymbol{\theta}_k, \sigma_k^2\big)\, P(\boldsymbol{\theta}_k|\,\lambda_{\theta_k}) \times \\ &\times\; P\big(\boldsymbol{y}^\star_{\cdot 1}|\,\xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2\big) \end{aligned} \tag{4.8}$$

which demands for a Metropolis-within-MCMC. After some mathematical steps, it is possible to show that, in the previous posterior distribution, the full conditionals of the parameters in $\boldsymbol{\Theta}$ are not in closed form. Our primary focus, however, is on the estimation of the $\boldsymbol{\beta}$ vector that contains the parameters truly describing the dynamic evolution of the ODEs' system. The spline smoothing step (Equation 4.2) is, instead, just a convenient approach to build the regression matrix $\tilde{H}$ in Equation 4.6. Moreover, an undesirable loop-feedback effect from $\boldsymbol{\beta}$ to $\boldsymbol{\Theta}$ (and viceversa) may arise if we aim for the 'true' full conditional obtained from Equation 4.8. So, in order to have a more stable and faster MCMC scheme, we adopt a Bayesian Smooth-and-Match strategy. The procedure consists of two steps:

- first (*smooth* step), we do Bayesian penalized spline smoothing to recover the $\boldsymbol{x}_{\cdot k}$'s through a Gibbs sampler that we obtain for $\boldsymbol{\theta}_k$ by temporarily not considering the integral solution term;
- second (*match* step), we plug-in the $\boldsymbol{x}_{\cdot k}$'s, computed with the sampled values for $\boldsymbol{\theta}_k$ from the previous step, into the sampler for $\boldsymbol{\beta}$ through the matrix $\tilde{H}$.

In practice, when sampling each $\boldsymbol{\theta}_k$, this allows us to drop the term $P\left(\boldsymbol{y}^{\star}_{\cdot 1} \,|\, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2\right)$. A more sophisticated smoothing could be performed (see Morrissey et al. [22] for a state-of-art Bayesian spline regression) but we consider a simpler approach because we are only interested, at the first step, in recovering the components of the process to be used as regressors later at the second step of the procedure. Notice that, from a frequentist point of view, a consistent estimator of the ODEs' parameters can be obtained under mild conditions (namely, on the penalizing terms) when following this plug-in approach [15]. The two steps of the strategy are not completely stand-alone compartments: as previously said, the quantity $\sigma_1^2$ connects both parts and acts as a measure for the uncertainty (from the smoothing and the regression) of the solution indirectly obtained by doing inference on $\boldsymbol{\beta}$. The other parameters of the model are all updated with Gibbs samplers that follow from standard derivations for conjugated priors and likelihood terms.

## 4.3 Simulation studies

In this Section, we validate our proposed strategy by testing it with three different ODEs' systems, starting from a simple one component model (logistic growth) up to a three components epidemic model (HIV viral fitness). We simulate nine scenarios for each ODEs model, exploring three different level of noise and three sample sizes ($n = 25, n = 100, n = 500$). The level of contamination of the data (low, medium or high), with Gaussian noise, is quantified through signal-to-noise ratio (SNR), that is the ratio between the standard deviation of the deterministic simulated solution (signal) and the standard deviation of the error term (noise) we use to perturb it. An increase in the strength of noise is equivalent to a decrease of the associated SNR, as the standard deviation of the signal at the numerator is fixed for a given sample size. For comparative purposes, we also inspect our

Bayesian Smooth-and-Match strategy (*SnM*) together with the collocation method implemented in the R package *CollocInfer* [24]. For each scenario, we run both algorithms on 100 independently simulated datasets and we summarize the results as the average across these replicates. We quantify the uncertainty about the estimated parameters in the same way for both algorithms by computing the associated mean square error (MSE) of these one hundred values. The tuning parameters for the collocation method, such as the number of knots, order of the basis for the splines and the penalization term, are selected with the functions provided in the package. The collocation method also needs initial guesses for the regression coefficients and we provide starting points drawn randomly from uniform distributions over ranges $[\boldsymbol{\beta} - 4\boldsymbol{\beta}; \boldsymbol{\beta} + 4\boldsymbol{\beta}]$; when the lower bound of the range is negative for parameters that only exist as positive we set it to a small value, close to zero. Given the sensibility of *CollocInfer* to the provided starting points, we control if convergence is achieved by the algorithm and we discard datasets that result in degenerate estimates for the parameters by checking the associated likelihood in the output. Those datasets are not considered when computing the averages and a measure of the number of actual datasets (NAD) used is provided. As for the visual representation of the results, Figures 4.2 to 4.4 show the plot mosaics with the results for one randomly drawn dataset of the simulated one hundred.

### 4.3.1 Logistic population growth

We first focus on a simple ODEs' system. We simulate data coming from the logistic growth model [21], frequently employed in ecology and biology to describe the growth dynamics of a certain population. The system is defined as:

$$\frac{d\boldsymbol{x}_{\cdot 1}}{dt} = a\boldsymbol{x}_{\cdot 1}\left(1 - \frac{\boldsymbol{x}_{\cdot 1}}{K}\right),$$

where $a$ is the growth rate and $K$ the carrying capacity of the population involved. We consider another representation of previous equation, that is, in our notation,

$$\frac{d\boldsymbol{x}_{\cdot 1}}{dt} = \beta_1\boldsymbol{x}_{\cdot 1} + \beta_2\boldsymbol{x}_{\cdot 1}^2,$$

linear in the parameters ($\beta_1 = a, \beta_2 = -a/K$) we want to estimate, together with $x_{11} = \xi_1$. We simulate a noise-free solution of the system and then we perturb it with Gaussian errors at different level of noise (low, medium, high); the three levels are obtained by setting the standard deviation of the errors as increasing proportions of the mean value of $\boldsymbol{x}_{\cdot 1}$. The number of knots for the smoothing step, $q_1 = 2$, is manually selected, aiming for some undersmoothing of $\boldsymbol{y}_{\cdot 1}$ [15]. The functions used to build the regression matrix are $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}^2$; the true values for $\xi_1$, $\beta_1$ and $\beta_2$ used in the simulations are reported in Table 4.1. As expected, we can see from Table 4.1 that to increasing levels of contamination of the data with noise correspond higher (on average) mean square errors for both methods. The average posterior mean of $\xi_1$ is stable throughout all the scenarios, showing some bias only when the SNR is at the highest level; *CollocInfer* does not provide an estimate for the initial condition $\xi_1$. With *SnM* we can recover the first parameter $\beta_1$ in almost any scenario with appreciable quality, also showing better results in comparison with *CollocInfer*; when the noise contamination is at its maximum, however, the MSE computed on the one hundred posterior means is noticeably higher. The algorithm *CollocInfer* seems to be more stable when retrieving the second parameter $\beta_2$, providing optimal estimates when the sample size is $n = 100$ and the noise contamination up to a medium level ($SNR = 13$ and $SNR = 6.5$). The solution uncertainty quantified by *SnM*, $\sigma_1^2$, appears to be less sensible to changes in sample size and more to the signal-to-noise ratio. In every scenario, the number of actual datasets (NAD) used to compute the results for *CollocInfer* is less than 100, meaning that degenerate solutions were discarded in the process. A visual description of the results is presented in Figure 4.2: in most of the plots, the line describing the true curve (*solid* line), the smoothed version of $\boldsymbol{y}_{\cdot 1}$ (*dotted* line) and the ODE regression solution (*long-dashed*) are undistinguishable from each other. They start to become appreciably different in the right part of the plots mosaic, showing the scenarios with the highest level of noise. We compute the average MSE between each curve (dotted line, $\text{MSE}_{x_1}$, and long-dashed line, $\text{MSE}_{g_1}$) and the true one representing the unperturbed data. For $n = 500$ and $SNR = 1.3$, the two mean square errors are $\text{MSE}_{x_1} = 0.022$, for the smoothed reconstruction, and $\text{MSE}_{x_1} = 0.007$ for the ODEs' system

solution; with the same sample size but less noise, SNR=13, the difference between the two curves and the true one is almost negligible ($\text{MSE}_{x_1} = 0.00015$ and $\text{MSE}_{g_1} = 0.00016$).

Table 4.1: Average posterior means (with *MSE* in brackets) for the parameters of the logistic population growth model; three sample sizes and increasing noise level

| Sample size | Parameters | Noise level | | | | | |
|---|---|---|---|---|---|---|---|
| | | low ($SNR = 13$) | | medium ($SNR = 6.5$) | | high ($SNR = 1.3$) | |
| | | SnM | CollocInfer | SnM | CollocInfer | SnM | CollocInfer |
| $n = 25$ | $\xi_1 = 0.1$ | 0.098 (*0.001*) | - | 0.096 (*0.002*) | - | 0.078 (*0.043*) | - |
| | $\beta_1 = 2.5$ | 2.531 (*0.105*) | 3.819 (*6.845*) | 2.560 (*0.412*) | 5.054 (*14.530*) | 2.317 (*4.687*) | 4.727 (*12.370*) |
| | $\beta_2 = -0.125$ | -0.180 (*0.271*) | -0.244 (*0.091*) | -0.216 (*1.035*) | -0.420 (*0.228*) | -0.359 (*9.864*) | -0.385 (*0.215*) |
| | $^{\diamond}\sigma_1^2$ | 0.064 | - | 0.244 | - | 7.054 | - |
| | NAD | 100 | 98 | 100 | 96 | 100 | 93 |
| $n = 100$ | $\xi_1 = 0.1$ | 0.098 (*0.001*) | - | 0.096 (*0.002*) | - | 0.078 (*0.046*) | - |
| | $\beta_1 = 2.5$ | 2.540 (*0.092*) | 5.080 (*15.480*) | 2.571 (*0.364*) | 5.060 (*15.267*) | 2.904 (*9.403*) | 5.017 (*14.886*) |
| | $\beta_2 = -0.125$ | -0.193 (*0.213*) | -0.124 (*0.009*) | -0.238 (*0.836*) | -0.124 (*0.009*) | -0.740 (*20.655*) | -0.130 (*0.011*) |
| | $^{\diamond}\sigma_1^2$ | 0.065 | - | 0.245 | - | 6.364 | - |
| | NAD | 100 | 92 | 100 | 92 | 100 | 94 |
| $n = 500$ | $\xi_1 = 0.1$ | 0.098 (*0.001*) | - | 0.095 (*0.002*) | - | 0.077 (*0.048*) | - |
| | $\beta_1 = 2.5$ | 2.542 (*0.086*) | 2.633 (*1.840*) | 2.573 (*0.342*) | 4.336 (*6.462*) | 2.852 (*8.649*) | 5.441 (*14.641*) |
| | $\beta_2 = -0.125$ | -0.197 (*0.188*) | -0.091 (*0.046*) | -0.242 (*0.742*) | -0.103 (*0.003*) | -0.662 (*18.647*) | -0.095 (*0.008*) |
| | $^{\diamond}\sigma_1^2$ | 0.066 | - | 0.248 | - | 6.126 | - |
| | NAD | 100 | 83 | 100 | 80 | 100 | 89 |

$\diamond$ results reported as multiplied by $10^2$

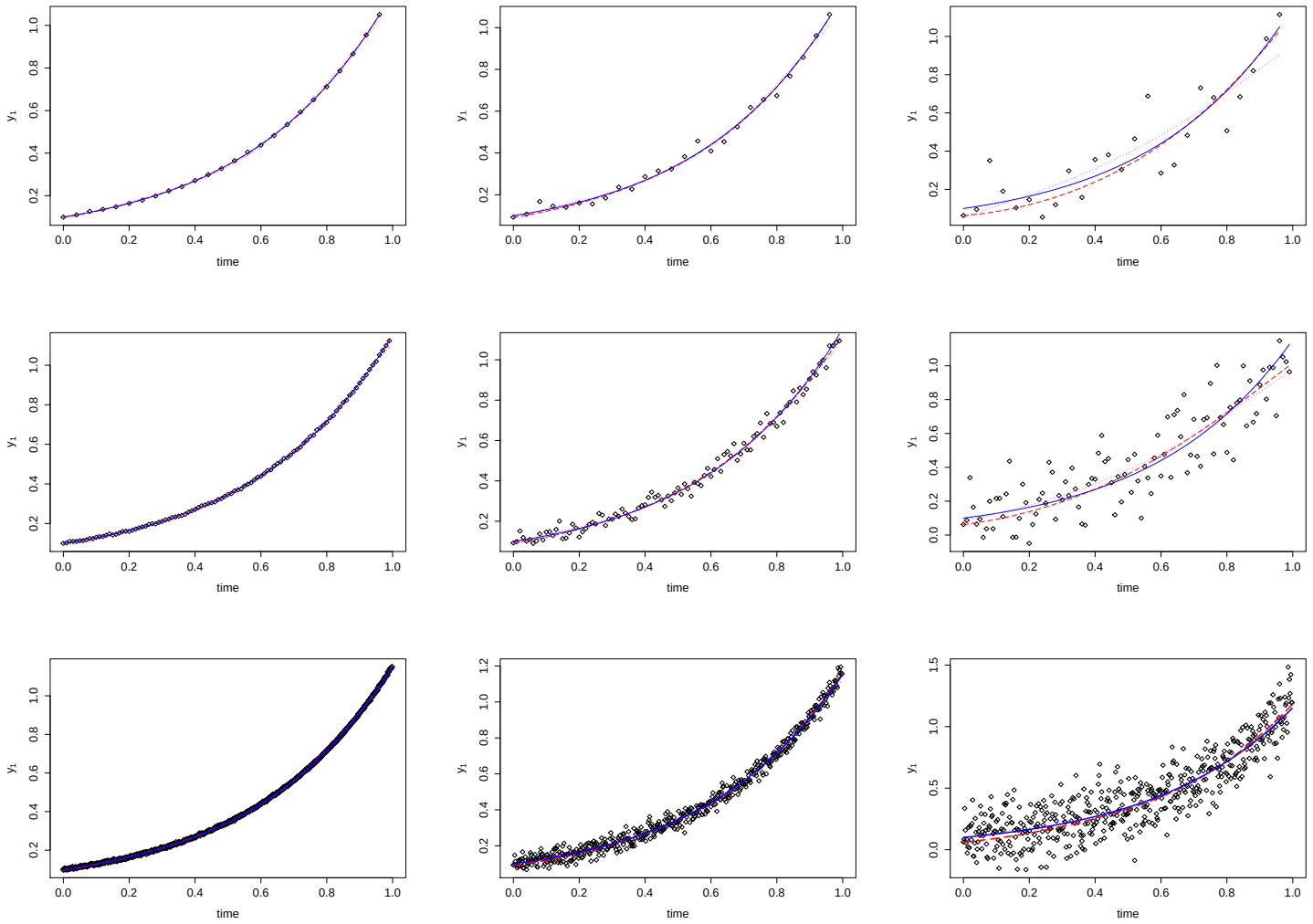


Figure 4.2: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

### 4.3.2 Lotka-Volterra

In the second batch of simulations we consider the Lotka-Volterra system [11]. This system of ODEs is used to model the dynamics, with respect to time, of two competing groups categorizable as *preys* and *predators*; setting some of the parameters of the ODEs to zero or imposing constraints, however, produces systems that are also used to characterize epidemic processes. The model is described by the following set of equations:

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_1 \boldsymbol{x}_{\cdot 1} + \beta_2 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 2} \\ \boldsymbol{x}'_{\cdot 2} = \beta_3 \boldsymbol{x}_{\cdot 2} + \beta_4 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 2} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2. \end{cases} \tag{4.9}$$

We focus the inference procedure on the first ordinary differential equation of the system and thus on the subset of parameters $(\xi_1, \beta_1, \beta_2)$. We explore nine scenarios as in the logistic growth case. We use the same approach to identify the three levels of noise but we select the sample sizes differently: the first one ($n = 25$) reproduces a situation where we have 25 evenly-spaced time points starting from $t_1 = 0$ up to $t_{25} = 24$; for $n = 100$, $t$ ranges from $t_1 = 0$ to $t_{100} = 99$ with unitary step size; the last one ($n = 500$) has the same time range as $n = 100$, with $t_1 = 0$ and $t_{500} = 99$, but a denser sampling grid given by 0.2 as the step size. Going from $n = 100$ to $n = 500$ encompass a situation where the time range is fixed (the maximum observational time point) but the amount of data increases (more observations in the same timeframe). We use as regressing functions the quantities $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 2}$; we choose the same number of knots, $q_1 = q_2 = 5$, for both splines. In Table 4.2, averages of the posterior means and corresponding mean square errors are reported for all the scenarios. We see that both algorithms perform well when the sample size is $n = 25$, regardless of the three noise level (SNR=50,5,2.5). The number of actual datasets used to compute averages for *CollocInfer* also shows that convergence was achieved for all the first three scenarios. When evaluating the performance of the two methods, for $n = 100$, we notice that *SnM* performs slightly better in terms of bias of the estimated parameters $\beta_1$ and $\beta_2$; also, *CollocInfer* shows slightly higher MSEs for

the second estimated parameter with respect to *SnM*. This difference is more prominent when $n = 500$. As expected, the quantification of the uncertainty about the solution provided by *SnM*, through $\sigma_1^2$, decreases as the sample size grows. A visual appreciation of the results is given in Figure 4.3. The most different performance between the smoothing and the regression curves is evident in the case of $n = 25$ and $SNR = 2.5$ (upper row of the mosaic, rightmost plot): in this case, the mean square errors of the dotted line and the long-dashed line are respectively $\text{MSE}_{x_1} = 0.690$ and $\text{MSE}_{g_1} = 0.252$. Such a discrepancy in the (average) accuracy of the reconstructed curves fades as the sample size grows. For $n = 500$ and $SNR = 50$ (the lowest noise level), the two errors are $\text{MSE}_{x_1} = 0.094$ and $\text{MSE}_{g_1} = 0.084$.

Table 4.2: Average posterior means (with *MSE* in brackets) for the parameters of the Lotka-Volterra ODEs' system; three sample sizes and increasing noise level

| Sample size | Parameters | Noise level | | | | | |
|---|---|---|---|---|---|---|---|
| | | low ($SNR = 50$) | | medium ($SNR = 5$) | | high ($SNR = 2.5$) | |
| | | SnM | CollocInfer | SnM | CollocInfer | SnM | CollocInfer |
| $n = 25$ | $\xi_1 = 2.0$ | 1.996 (*0.001*) | - | 1.965 (*0.112*) | - | 1.929 (*0.448*) | - |
| | $\beta_1 = 0.1$ | 0.101 (*0.001*) | 0.095 (*0.001*) | 0.101 (*0.001*) | 0.093 (*0.001*) | 0.091 (*0.001*) | 0.863 (*0.001*) |
| | $\beta_2 = -0.2$ | -0.201 (*0.001*) | -0.197 (*0.001*) | -0.197 (*0.001*) | -0.192 (*0.002*) | -0.170 (*0.002*) | -0.183 (*0.008*) |
| | $^{\diamond}\sigma_1^2$ | 0.040 | - | 2.295 | - | 12.234 | - |
| | NAD | 100 | 100 | 100 | 100 | 100 | 100 |
| $n = 100$ | $\xi_1 = 2.0$ | 1.996 (*0.001*) | - | 1.965 (*0.112*) | - | 1.930 (*0.449*) | - |
| | $\beta_1 = 0.1$ | 0.089 (*0.001*) | 0.063 (*0.001*) | 0.088 (*0.001*) | 0.060 (*0.002*) | 0.086 (*0.001*) | 0.061 (*0.002*) |
| | $\beta_2 = -0.2$ | -0.168 (*0.001*) | -0.140 (*0.004*) | -0.167 (*0.001*) | -0.139 (*0.004*) | -0.162 (*0.002*) | -0.138 (*0.004*) |
| | $^{\diamond}\sigma_1^2$ | 3.650 | - | 5.392 | - | 10.507 | - |
| | NAD | 100 | 100 | 100 | 100 | 100 | 98 |
| $n = 500$ | $\xi_1 = 2.0$ | 1.996 (*0.001*) | - | 1.965 (*0.112*) | - | 1.930 (*0.447*) | - |
| | $\beta_1 = 0.1$ | 0.095 (*0.001*) | 0.080 (*0.001*) | 0.095 (*0.001*) | 0.080 (*0.001*) | 0.094 (*0.002*) | 0.077 (*0.001*) |
| | $\beta_2 = -0.2$ | -0.182 (*0.001*) | -0.138 (*0.004*) | -0.182 (*0.001*) | -0.138 (*0.004*) | -0.180 (*0.002*) | -0.138 (*0.004*) |
| | $^{\diamond}\sigma_1^2$ | 1.405 | - | 2.89 | - | 7.407 | - |
| | NAD | 100 | 100 | 100 | 100 | 100 | 99 |

$\diamond$ results reported as multiplied by $10^1$

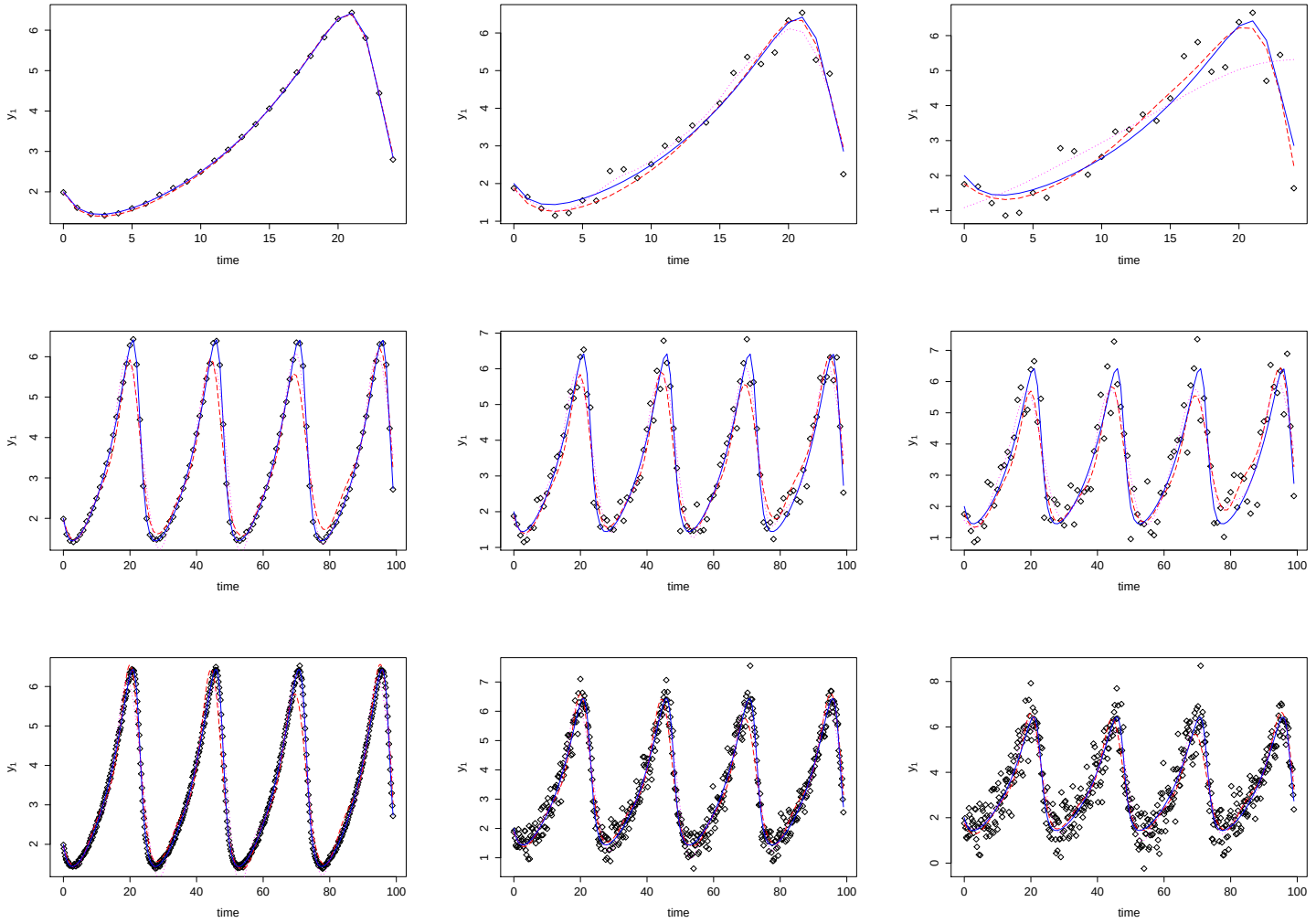


Figure 4.3: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

### 4.3.3 HIV viral fitness

Our last batch of data is simulated from a set of ODEs modelling the dynamics of HIV virus [3]. The system is defined as:

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_1 + \beta_2 \boldsymbol{x}_{\cdot 1} + \beta_3 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3} \\ \boldsymbol{x}'_{\cdot 2} = \beta_3 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3} - 0.5 \boldsymbol{x}_{\cdot 2} \\ \boldsymbol{x}'_{\cdot 3} = 0.5 \cdot \beta_4 \boldsymbol{x}_{\cdot 2} + \beta_5 \boldsymbol{x}_{\cdot 3} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \\ x_{13} = \xi_3. \end{cases} \tag{4.10}$$

where we focus on the first ODE and on the subset of parameters $(\xi_1, \beta_1, \beta_2, \beta_3)$. The simulation setting is the same as in Section 4.3.2: three levels of noise and three sample sizes where only the last has an effective increase in the number of observations. True parameters' values are tuned according to the ones used in Vujačić et al. [28]. Number of knots selected is the same for all splines $q_1 = q_2 = q_3 = 20$. We use the following regressing functions: $h_1(\boldsymbol{x}) = 1$, $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_3(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3}$. We report the results in Table 4.3 and Figure 4.4. The parameter $\beta_2$ proved to be difficult to estimate with *CollocInfer* so we decided to fix it to the true value for this algorithm while estimating it in the case of *SnM*. When $n = 25$ we notice that our method breaks down at the highest level of noise contamination (SNR=1.3): set aside the recovered initial condition, which shows good average posterior mean, the estimates for the other parameters are noticeably biased and have large mean square errors (especially in the case of $\beta_1$). Even if the NAD for *CollocInfer* is around 85 for the first three scenarios, the algorithm seems to be more robust in this setting. When considering $\beta_1$ and $\beta_3$, *CollocInfer* achieves lower mean square errors in almost every scenario with the exception of the setting with $n = 500$ and SNR=15, where the MSE for $\beta_1$ obtained with *SnM* is slightly lower than the one computed for *CollocInfer*. The number of discarded datasets gets lower as the sample size grows, as expected. For $n = 100$ and $n = 500$ we do not observe the same degrading effect on our algorithm *SnM* even at the highest level of noise, meaning that the information from the bigger sample size is enough to overcome the contamination in the data. As we can see from the mosaic plot in Figure 4.4, for $n = 100$ and

$n = 500$ there is an appreciable difference between the two curves with respect to the true one, especially on the right-half portion of each plot. The average mean square errors for the setting with $n = 100$ and SNR=1.3 are $\text{MSE}_{g_1} = 16.179$ and $\text{MSE}_{x_1} = 11.743$, meaning that the smoothing step provides a reconstructed curve slightly closer to the true one (dotted line versus solid line). The opposite happens when the sample size gets to 500, as in the setting with SNR=2.5, the two MSEs are and $\text{MSE}_{g_1} = 1.397$ and $\text{MSE}_{x_1} = 5.859$. As for the setting in which the model breaks down ($n = 25$, SNR=1.3), the average mean square errors are $\text{MSE}_{g_1} = 80.467$ and $\text{MSE}_{x_1} = 104.969$, values much higher than the ones computed on the other eight scenarios (not reported here for brevity).

Table 4.3: Average posterior means (with *MSE* in brackets) for the parameters of the HIV viral fitness ODEs' system; three sample sizes and increasing noise level

| Sample size | Parameters | Noise level | | | | | |
|---|---|---|---|---|---|---|---|
| | | low ($SNR = 15$) | | medium ($SNR = 2.5$) | | high ($SNR = 1.3$) | |
| | | SnM | CollocInfer | SnM | CollocInfer | SnM | CollocInfer |
| $n = 25$ | $\xi_1 = 60$ | 59.879 (*1.35*) | - | 59.519 (*21.55*) | - | 59.031 (*86.12*) | - |
| | $\beta_1 = 20$ | 20.808 (*2.79*) | 20.933 (*1.001*) | 20.294 (*21.89*) | 21.142 (*7.130*) | 14.609 (*175.49*) | 20.849 (*8.869*) |
| | $\beta_2 = -0.108$ | -0.113 (*0.001*) | - | -0.103 (*0.005*) | - | -0.030 (*0.032*) | - |
| | $^{\dagger}\beta_3 = -0.095$ | -0.106 (*0.001*) | -0.106 (*0.003*) | -0.109 (*0.001*) | -0.109 (*0.001*) | -0.159 (*0.001*) | -0.106 (*0.001*) |
| | $^{\diamond}\sigma_1^2$ | 0.169 | - | 2.346 | - | 20.398 | - |
| | NAD | 100 | 84 | 100 | 85 | 100 | 85 |
| $n = 100$ | $\xi_1 = 60$ | 59.882 (*1.18*) | - | 59.539 (*18.82*) | - | 59.079 (*75.26*) | - |
| | $\beta_1 = 20$ | 21.714 (*3.13*) | 19.700 (*1.936*) | 21.457 (*4.98*) | 19.634 (*2.188*) | 20.281 (*9.94*) | 19.772 (*3.728*) |
| | $\beta_2 = -0.108$ | -0.117 (*0.001*) | - | -0.115 (*0.001*) | - | -0.106 (*0.001*) | - |
| | $^{\dagger}\beta_3 = -0.095$ | -0.103 (*0.001*) | -0.090 (*0.001*) | -0.103 (*0.001*) | -0.089 (*0.010*) | -0.100 (*0.001*) | -0.090 (*0.009*) |
| | $^{\diamond}\sigma_1^2$ | 0.640 | - | 2.748 | - | 9.473 | - |
| | NAD | 100 | 98 | 100 | 98 | 100 | 97 |
| $n = 500$ | $\xi_1 = 60$ | 59.882 (*1.18*) | - | 59.538 (*18.84*) | - | 59.078 (*75.35*) | - |
| | $\beta_1 = 20$ | 21.012 (*1.16*) | 19.612 (*1.791*) | 21.040 (*3.17*) | 19.648 (*1.412*) | 20.936 (*8.99*) | 19.499 (*2.437*) |
| | $\beta_2 = -0.108$ | -0.114 (*0.001*) | - | -0.113 (*0.001*) | - | -0.112 (*0.001*) | - |
| | $^{\dagger}\beta_3 = -0.095$ | -0.100 (*0.001*) | -0.088 (*0.001*) | -0.100 (*0.001*) | -0.089 (*0.001*) | -0.101 (*0.001*) | -0.087 (*0.001*) |
| | $^{\diamond}\sigma_1^2$ | 0.437 | - | 2.312 | - | 8.326 | - |
| | NAD | 100 | 99 | 100 | 99 | 100 | 100 |

† results reported as multiplied by $10^2$

◇ results reported as multiplied by $10^{(-1)}$

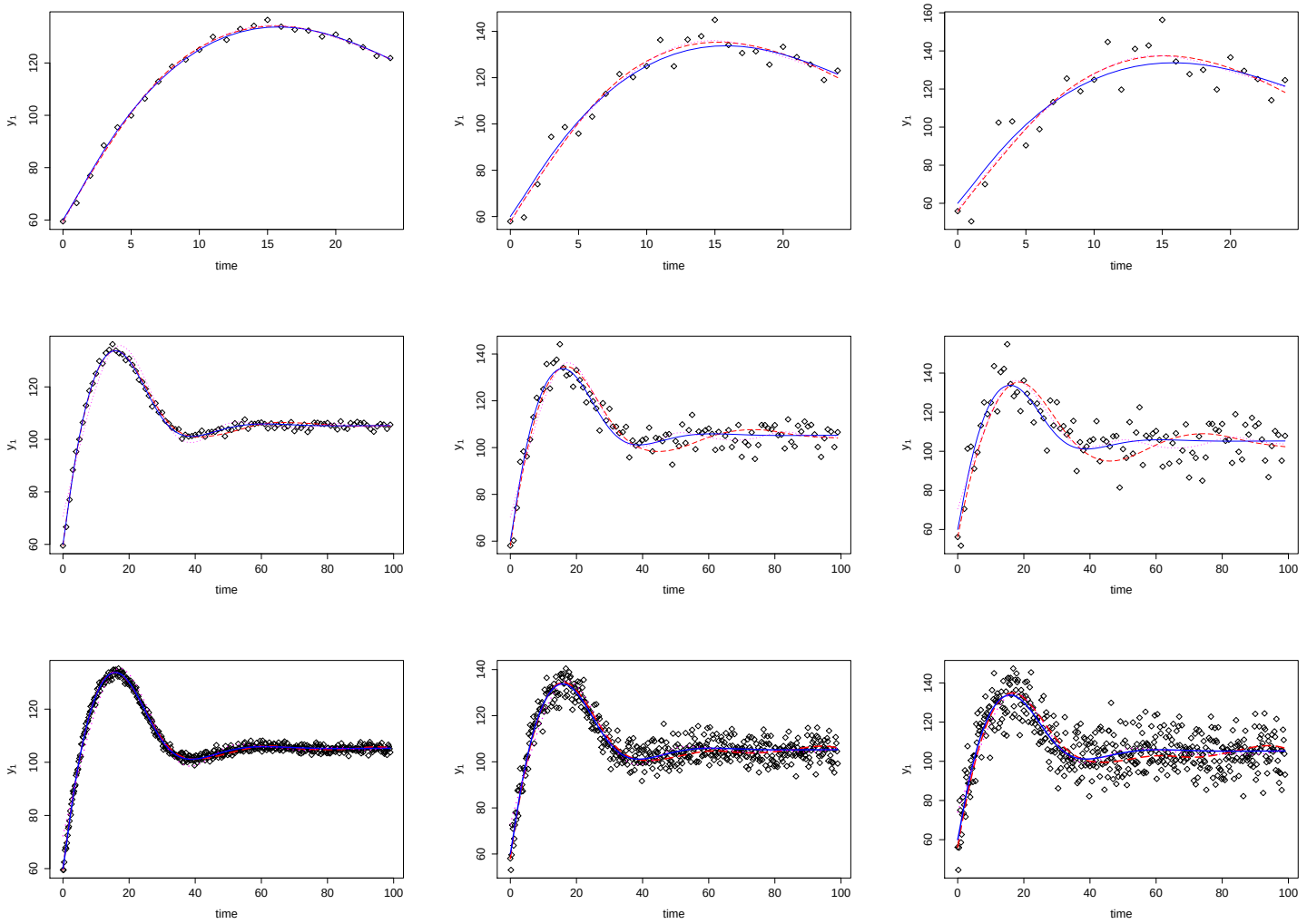


Figure 4.4: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

### 4.3.4 FitzHugh-Nagumo system for neuron electrical activity

We analyze a toy-data example available from the package *CollocInfer* [17]. The data (*FhNdata*) consist of 41 evenly-spaced observations in the timeframe $[0, 20]$ from the following ODEs model

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = c(\boldsymbol{x}_{\cdot 1} - \boldsymbol{x}^3_{\cdot 1}/3 + \boldsymbol{x}_{\cdot 2}) \\ \boldsymbol{x}'_{\cdot 2} = -\frac{1}{c}(\boldsymbol{x}_{\cdot 1} - a + b\boldsymbol{x}_{\cdot 2}) \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \end{cases} \tag{4.11}$$

known as the FitzHugh-Nagumo system [12, 23]. The set of equations describe the pulse transmission for neuronal activity. The parameters' values used to generate the data are $a = 0.2, d = 0.2, c = 3, \xi_1 = 0.5$; the simulated values are then perturbed with variances equal to 0.25 for both the variables $\boldsymbol{x}_{\cdot 1}$ and $\boldsymbol{x}_{\cdot 2}$. As it is written in Equation 4.11, the system is not linear in the parameters. We thus consider another representation

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_4(\boldsymbol{x}_{\cdot 1} - \boldsymbol{x}^3_{\cdot 1}/3 + \boldsymbol{x}_{\cdot 2}) \\ \boldsymbol{x}'_{\cdot 2} = \beta_1\boldsymbol{x}_{\cdot 1} + \beta_2 + \beta_3\boldsymbol{x}_{\cdot 2} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \end{cases} \tag{4.12}$$

and we focus on the second differential equation and the subset of parameters $(\xi_2, \beta_1 = -1/c, \beta_2 = a/c, \beta_3 = -d/c)$. The ODE solution regression uses the following functions: $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$, $h_2(\boldsymbol{x}) = 1$ and $h_3(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 2}$. We compare our results with the point estimates obtained by Vujačić et al. [28], remarking the fact that we only do inference on one of the two differential equations whereas they simultaneously estimate all the parameters of the model using information from both variables. In Figure 4.5, reconstructed curves from both steps of the procedure (*smooth* and *match*) are plotted: we can see, in the right plot, the penalized spline (*dotted* line) captures a biased initial condition in comparison to the ODE solution regression (*long-dashed* line), even if the associated mean square errors for the two curves, respectively $\text{MSE}_{g_1} = 0.042$ and $\text{MSE}_{x_1} = 0.033$, are actually close. The degree of smoothing for the two curves is also practically the same. The

smoothing for the first variable (left plot of Figure 4.5) is satisfactory. As for the other parameters, we report their posterior means in Table 4.4: the algorithm recovers the true value for $\beta_1$ with appreciable accuracy; for $\beta_2$ and especially $\beta_3$, the algorithm returns slightly biased posterior means. Although, as stated before, there is no theoretical true correspondence between the estimated shared nuisance parameter and the variance of the noise added to the data, the posterior mean for $\sigma_2^2$ is lower than the one used to perturb the data, meaning that - potentially - the additional information from the ODE solution helps shrinking down the overall measure of uncertainty regarding the second variable $\boldsymbol{x}_{\cdot 2}$ that we are trying to model.

Table 4.4: Results for the *FhNdata*: posterior means (*posterior standard deviations* within brackets) from *SnM* in the second column and point estimates from Vujačić et al. [28] in the third column

| Parameters | Post. Mean (*post. sd*) | Vujačić et al. [28] |
|---|---|---|
| $\xi_2 = 0.5$ | 0.696 (*0.244*) | 0.569 |
| $\beta_1 = -0.33$ | -0.322 (*0.041*) | -0.333 |
| $\beta_2 = 0.067$ | 0.091 (*0.025*) | 0.106 |
| $\beta_3 = -0.067$ | -0.028 (*0.066*) | -0.047 |
| $\sigma_2^2$ | 0.060 (*0.019*) | |

Runtime: 10,000 MCMC iterations in 29.17 seconds.

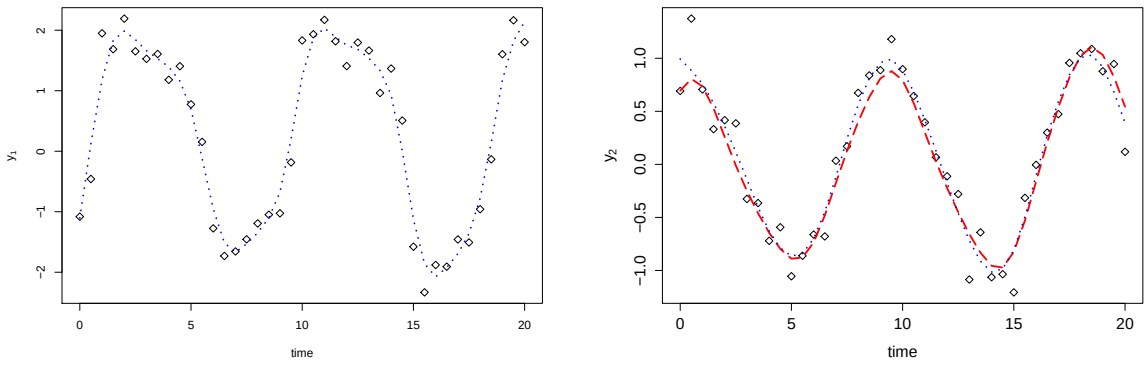


Figure 4.5: Observed noisy data (*dots*), smoothing spline (*dotted* line) and reconstructed solution (*long-dashed* line, only right plot) for the first variable (left plot) and the second variable $\boldsymbol{x}_{\cdot 2}$ (right plot)

## 4.4 Conclusions

We have proposed a Bayesian approach to indirectly solve an equation of an ODEs' system while doing inference on its parameters. The employed strategy is compartmentalized into two main stages: a first *smoothing* step, that serves as a reconstruction of the components of the process through penalized spline smoothing of the noisy observed data; a second *match* step, where the smoothed curves are numerically integrated and used as inputs for ridge regression. The two phases of the procedure are jointly governed by $\sigma_k^2$, a noise parameter - common to both steps - that measures the solution uncertainty of the $k$-th equation of the system we are indirectly solving. We evaluated the performance and reliability of the strategy through different ODEs' systems, starting from a simple one (with only one variable) and then moving to processes that had two or three variables and more complex time dynamics. We also tested the approach on a dataset previously analyzed by [28], to compare the results. The procedure we propose has the advantages of being fast, with a built-in quantification of uncertainty about the ODE solution obtained as a by-product of the inference. The 'tuning' parameters are minimal: the number of knots and their placement have no substantial impact on the reliability of the smoothing step as long as the choice is reasonable with respect to the problem we are dealing with; different splines (i.e, *B-spline*, *thin plates*, etc.), that do not actually require such a choice, can be employed in the first step to address the issue. As far as the integration is concerned, we rely on an easy to implement - albeit 'rough' - trapezoidal rule that uses the observed time points as the grid to evaluate the integral: a better approximation can be achieved by employing a finer grid, at the cost of increased computational times. Other types of penalization, instead of the ridge, could be explored for the regression step of the strategy.

For future developments, we aim to be able to estimate all the parameters of the system while indirectly solving together and simultaneously the ordinary differential equations, instead of focusing on one of them. As trivial as it may seems to extend the approach, careful consideration is needed before moving toward this direction. For example, a first idea would be to independently run the procedure for each equation but, in that case, we would not be truly using the information at our disposal about the relationships between the variables.

Another issue would be, when considering all the equations together, which of the two reconstructed curves (the smoothed spline and the regression solution) to use at each iteration of the MCMC procedure given that the measure of uncertainty considers them both. An interesting extension of the model could consider the regressing functions $h_j(\boldsymbol{x})$ as not known in advance and to be estimated along with the other parameters of the model. We already briefly explored this aspect using a Bayesian smoothed spline regression at the *match* step of the procedure: we obtained a satisfactory reconstruction of the curve but at the expense of losing interpretability of the parameter vector $\boldsymbol{\beta}$, meaning that further investigation on this topic is needed. Finally, as seen in Section 4.2.3, an interesting aspect would be to investigate deeper (and potentially quantify, theoretically) the consequences of the shared nuisance parameter $\sigma_1^2$. In this direction, an option could be to express the two levels of noise from the two sources of information ($\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}_1^\star$) as a proportion of the total shared nuisance parameter and to decide which curve to use, the smoothed one or the system's solution, based on these fractions.

# Bibliography

[1] Barber, David. On solving Ordinary Differential Equations using Gaussian Processes. *arXiv preprint arXiv:1408.3807*, 2014.

[2] Barber, David and Wang, Yali. Gaussian Processes for Bayesian Estimation in Ordinary Differential Equations. In Jebara, Tony and Xing, Eric P., editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1485–1493. JMLR Workshop and Conference Proceedings, 2014. URL http://jmlr.org/proceedings/papers/v32/barber14.pdf.

[3] Bonhoeffer, Sebastian, May, Robert M, Shaw, George M, and Nowak, Martin A. Virus dynamics and drug therapy. *Proceedings of the National Academy of Sciences*, 94(13):6971–6976, 1997.

[4] Brunel, Nicolas JB et al. Parameter estimation of ODE's via nonparametric estimators. *Electronic Journal of Statistics*, 2:1242–1267, 2008.

[5] Calderhead, Ben, Girolami, Mark, and Lawrence, Neil D. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.

[6] Campbell, David and Steele, Russell J. Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22(2):429–443, 2012.

[7] Chkrebtii, Oksana A, Campbell, David A, Calderhead, Ben, and Girolami, Mark A. Bayesian Solution Uncertainty Quantification for Differential Equations. *arXiv preprint arXiv:1306.2365*, 2013.

[8] Conrad, Patrick R, Girolami, Mark, Särkkä, Simo, Stuart, Andrew, and Zygalakis, Konstantinos. Probability measures for numerical solutions of differential equations. *arXiv preprint arXiv:1506.04592*, 2015.

[9] Dattner, Itai and Klaassen, Chris AJ. Estimation in Systems of Ordinary Differential Equations Linear in the Parameters. *arXiv preprint arXiv:1305.4126*, 2013.

[10] Dondelinger, F., Filippone, M., Rogers, S., and Husmeier, D. ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013. URL http://eprints.gla.ac.uk/77857/.

[11] Edelstein-Keshet, Leah. *Mathematical models in biology*, volume 46. Siam, 1988.

[12] FitzHugh, Richard. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445, 1961.

[13] González, Javier, Vujačić, Ivan, and Wit, Ernst. Inferring latent gene regulatory network kinetics. *Statistical applications in genetics and molecular biology*, 12 (1):109–127, 2013.

[14] González, Javier, Vujačić, Ivan, and Wit, Ernst. Reproducing kernel Hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32, 2014.

[15] Gugushvili, Shota, Klaassen, Chris AJ, et al. $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, 18(3):1061–1098, 2012.

[16] Honkela, Antti, Girardot, Charles, Gustafson, E Hilary, Liu, Ya-Hsin, Furlong, Eileen EM, Lawrence, Neil D, and Rattray, Magnus. Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, 107(17):7793–7798, 2010.

[17] Hooker, G, Xiao, L, and Ramsay, J. Collocinfer: collocation inference for dynamic systems. *R package version 0.1.0*, 2010.

[18] Liang, Hua and Wu, Hulin. Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 2012.

[19] Macdonald, Benn, Higham, Catherine, and Husmeier, Dirk. Controversy in mechanistic modelling with Gaussian processes. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 37, pages 1539–1547. Microtome Publishing, 2015.

[20] Madár, János, Abonyi, János, Roubos, Hans, and Szeifert, Ferenc. Incorporating Prior Knowledge in a Cubic Spline Approximation Application to the Identification of Reaction Kinetic Models. *Industrial & engineering chemistry research*, 42(17):4043–4049, 2003.

[21] McKendrick, A. G. and Pai, M. Kesava. The Rate of Multiplication of Microorganisms: A Mathematical Study. *Proceedings of the Royal Society of Edinburgh*, 31:649–653, 1 1912. ISSN 0370-1646. doi: 10.1017/S0370164600025426.

[22] Morrissey, Edward R, Juárez, Miguel A, Denby, Katherine J, and Burroughs, Nigel J. Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully Bayesian spline autoregression. *Biostatistics*, 12(4): 682–694, 2011.

[23] Nagumo, Jinichi, Arimoto, Suguru, and Yoshizawa, Shuji. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[24] Ramsay, Jim O, Hooker, G, Campbell, D, and Cao, J. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

[25] Robinson, James C. *An introduction to ordinary differential equations*. Cambridge University Press, 2004.

[26] Titsias, Michalis K, Honkela, Antti, Lawrence, Neil D, and Rattray, Magnus. Identifying targets of multiple co-regulating transcription factors from expression time-series by Bayesian model comparison. *BMC systems biology*, 6(1):1, 2012.

[27] Varah, JM. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46, 1982.

[28] Vujačić, Ivan, Dattner, Itai, González, Javier, and Wit, Ernst. Time-course window estimator for ordinary differential equations linear in the parameters. *Statistics and Computing*, 6:1057–1070, 2015.

[29] Wood, Simon. *Generalized additive models: an introduction with R*. CRC Press, 2006.

[30] Xue, Hongqi, Miao, Hongyu, and Wu, Hulin. Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. *Annals of statistics*, 38(4):2351, 2010.